Calhoun: The NPS Institutional Archive

DSpace Repository

Reports and Technical Reports All Technical Reports Collection

1994-09-30

Multicast tree construction in network topologies with asymmetric link loads

Shukla, Shridhar B.

Monterey, California. Naval Postgraduate School

http://hdl.handle.net/10945/28767

Downloaded from NPS Archive: Calhoun

DUDLEY KNOX LIBRARY

Calhoun is a project of the Dudley Knox Library at NPS, furthering the precepts and goals of open government and government transparency. All information contained herein has been approved for release by the NPS Public Affairs Officer.

Dudley Knox Library / Naval Postgraduate School
411 Dyer Road / 1 University Circle
Monterey, California USA 93943

http://www.nps.edu/library

NPS-EC-94-012

NAVAL POSTGRADUATE SCHOOL
Monterey, California

Multicast Tree Construction in Network Topologies with Asymmetric Link Loads

by

Shridhar B. Shukla
J. Eric Klinker
Eric B. Boyer

September 30, 1994

Approved for public release; distribution is unlimited.

FedDocs
D 208.14/2
NPS-EC-94-012

NPS-EC-94-012

Naval Postgraduate School

Monterey, California 93943-5000

Read Admiral T. A. Mercer — H. Shull
Superintendent — Provost

Shridhar Shukla was funded by the NSF RIA Grant 9309316

Approved for public release; distribution unlimited.

This report was prepared by:

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time reviewing instructions, searching existing data sources gathering and maintaining the data needed, and completing and reviewing the collection of information Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188), Washington, DC 20503.

DUDLEY KNOX LIBRARY
NAVAL POSTGRADUATE SCHOOL
MONTEREY CA 93943-5101

1. AGENCY USE ONLY (Leave Blank)

2. REPORT DATE
September 1994

3. REPORT TYPE AND DATES COVERED
Final Report

4. TITLE AND SUBTITLE
Multicast Tree Construction in Network Topologies with Asymmetric Link Loads

5. FUNDING NUMBERS

6. AUTHOR(S)
Shridhar B. Shukla, J. Eric Klinker, Eric B. Boyer

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Naval Postgraduate School
Monterey, CA 93943-5000

8. PERFORMING ORGANIZATION REPORT NUMBER
NPS-EC-94-012

9. SPONSORING/ MONITORING AGENCY NAME(S) AND ADDRESS(ES)

10. SPONSORING/ MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES
The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the United States Government.

12a. DISTRIBUTION / AVAILABILITY STATEMENT
Approved for public release; distribution is unlimited.

12b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 words)*

This report addresses the problem of constructing multicast trees with reservation of resources. The main features of the approach described are that it tolerates asymmetric traffic loads on network links and algorithmically locates data distribution centers for every multiparticipant interaction. A fast and scalable algorithm for locating distribution centers based on the network load and a priori knowledge of participant's locations and resource requirements is given. To explicitly handle cases of disjoint send and receive paths between two nodes, a protocol to build separate send-trees and receive-trees around the centers located in the manner above is given. Simulation results on various topologies are presented showing that, with the above center location mechanism, center-specific trees yield lower tree cost than source-specific trees for many concurrent senders without increasing the average path length significantly. The use of distribution centers, a priori information, and sensitivity to load asymmetry permit effective combination of center-specific and source-specific trees for an interaction and eliminate the need for symmetry checks during resource reservation.

14. SUBJECT TERMS
Multicast trees, scalable, quality of service, wide area networks

15. NUMBER OF PAGES

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT
Unclassified

18. SECURITY CLASSIFICATION OF THIS PAGE
Unclassified

19. SECURITY CLASSIFICATION OF ABSTRACT
Unclassified

20. LIMITATION OF ABSTRACT
Unlimited

MULTICAST TREE CONSTRUCTION IN NETWORK TOPOLOGIES WITH ASYMMETRIC LINK LOADS

Shridhar B. Shukla
Eric B. Boyer

Department of Electrical and Computer Engineering
Naval Postgraduate School
Monterey, CA 93943-5121

J. Eric Klinker

Center of High Assurance Computing Systems
Naval Research Laboratory
Washington, D.C. 20375-5000

September 30, 1994

ABSTRACT

This report addresses the problem of constructing multicast trees with reservation of resources. The main features of the approach described are that it tolerates asymmetric traffic loads on network links and algorithmically locates data distribution centers for every multiparticipant interaction. A fast and scalable algorithm for locating distribution centers based on the network load and *a priori* knowledge of participant's locations and resource requirements is given. To explicitly handle cases of disjoint send and receive paths between two nodes, a protocol to build separate send-trees and receive-trees around the centers located in the above manner is given. Simulation results on various topologies are presented showing that, with the above center location mechanism, center-specific trees yield lower tree cost than source-specific trees for many concurrent senders without increasing the average path length significantly. The use of distribution centers, *a priori* information, and sensitivity to load asymmetry permit effective combination of center-specific and source-specific trees for an interaction and eliminated the need for symmetry checks during resource reservation.

1. INTRODUCTION

1.1 BACKGROUND

The integrated packet switched networks of the future are expected to provide users with a variety of multiparty interaction capabilities. These services will benefit from a network level multicast with guaranteed Quality of Service (QoS). Guaranteed QoS, in terms of delay and jitter bounds, can be provided through the reservation of resources such as buffer and packet processing capacity at network nodes [10]. Network level multicast pioneered in [7, 8] refers to the reduction in the amount of traffic for point to multipoint (group) communication when compared with unicasts. One way to provide network level multicast requires the network to form a multicast routing tree based on the members of the group and their location [8].

There are two basic approaches to multicast tree construction. The first is a shared or center-specific tree (CST) [1] and the other is a source based or source-specific tree (SST) [5, 7]. A center-specific approach utilizes a single tree rooted at some center that is shared by all senders. In the source-specific approach each sender builds a separate tree rooted at itself. The center-specific tree reserves fewer resources for an interaction that contains multiple, yet non-concurrent, senders that require reservations. The disadvantages of a center-specific tree are that over large groups, certain links may become bottlenecks and in the case of a large number of concurrent senders, traffic concentration may occur [12]. Current techniques suggest that the center (core) be selected administratively. Ideally, it should be selected algorithmically based on the participants' locations and network load distribution.

The source-specific tree approach is scalable and efficient in interactions with a large number of concurrent senders. The volume carried along each tree is the same regardless of the number of senders. The source-specific tree makes excessive reservations when the number of concurrent senders is small compared to the total number of senders requiring guaranteed QoS. The reservations will occur along every tree although every tree does not carry traffic at the same time. Thus, depending upon the type of reservation-based interaction to be set up, either a center-specific tree or a source-specific tree will be most economical. However, neither approach implements tree construction according to the availability of the resources required to guarantee QoS. Since resources are consumed along the routes taken by the multicast traffic, an approach that provides guaranteed QoS should couple tree construction with resource availability. The current draft-standard for resource reservation being considered by the Internet Engineering Task Force (IETF) is the Resource ReSerVation Protocol (RSVP) [14]. RSVP proposes receiver-initiated reservation of resources which are completely independent of the way the network routes packets and creates multicast trees.

1.2 MOTIVATION AND OBJECTIVES

The motivation for this work is as follows. Multicasting with guaranteed QoS should permit a flexible choice of center-specific trees and source-specific trees based on the number and the nature of participants. The choice of tree type should not be left entirely up to the receiver but should be based, to the extent possible, on *a priori* information about the participants. The tree centers should be selected algorithmically based on this information. A resource availability check should be made at tree construction time to spread the load uniformly and to prevent cases of unobtainable reservations after the tree is set up. Finally, the burden of resource reservation should be shared by the senders and the receivers, particularly when the senders are numerous and relatively long-lived. Instead of making the receivers obtain reservations to every sender, such senders should obtain reservations up to some distribution center and the receivers should obtain reservations from the center.

While it is reasonable to expect that most of the links in the Internet are symmetric in their capacities, it may be unreasonable to expect the traffic load on any given link to be symmetric (consider the case of ftp, most of the bandwidth is consumed in only one direction). It is shown in [20, 21] that loads on the NSFNET backbone are not symmetric. A single-site study [22] confirmed the existence of certain "busy source" or "favorite site" effects, where a small number of hosts dominate the network traffic. These effects should contribute to network asymmetry. This is particularly true if the routing techniques used in the network do not support asymmetric topologies, since any degree of asymmetry is likely to be heightened by the routing technique. These results are supported by simulations where multiple interactions using a reverse path routing mechanism were built on a uniform topology and the resultant topology was not symmetric [15].

The specific objectives of this report are:

1. Define an approach for multicast data distribution that permits flexible construction of center-specific and source-specific trees using *a priori* information about participants.

2. Determine an algorithmic technique to locate a distribution center for a center-specific tree.

3. Describe an approach for center-specific tree construction in the presence network asymmetry. The approach should be valid for symmetric topologies.

4. Compare the quality of the resultant center-specific trees with sender-specific trees.

1.3 ORGANIZATION

The report is organized as follows. Section 2 describes our approach to meet the design goals listed above. Section 3 describes the protocols for center selection and tree construction required by the approach. Section 4 describes the performance evaluation of the approach. Section 5 details a comparison with some existing techniques. Section 6 concludes with general results and directions for future research.

2. GENERAL APPROACH

2.1 USE OF *A PRIORI* INFORMATION ABOUT THE PARTICIPANTS

The requirements of a multiparty interaction are determined by the following three characteristics: reservation requirements of individual participants, the number of concurrent senders, and the location of the participants. Thus, every participant should know an estimate of its resource requirements (bandwidth), its address (location), and the nature of the role it will play in the interaction (specifically its sending/receiving requirements). We anticipate a network entity called a *scheduler*, similar to the session directory (sd) [16] tool. The scheduler can determine the requirements of the interaction if it is given the above information about potential participants. The scheduler is responsible for grouping participants into groups referred to as *critical sets of participants* (CSP). From the sending requirements of a participant, the scheduler can determine if that participant should form a source-specific tree or join a shared center-specific tree. If the participant is expected to source traffic throughout the interaction then a source-specific tree is most efficient. On the other hand, if the participant is a member of a group that is expected to have a single sender at any give time, then those participants should share a tree. Participants in distant and separate routing domains should not share a tree. Thus, based on *a priori* information, the combination of source-specific trees and center-specific trees for an interaction can be controlled.

If center-specific trees are required, the router that acts as the center must be determined. The same *a priori* information can be used to select the location of a center that best fits the needs of the interaction. The following section details an approach that locates the distribution centers of an interaction based only on the *a priori* information listed above.

2.2 ALGORITHMIC LOCATION OF A SET OF DISTRIBUTION CENTERS

In this approach, there is one distribution center located for each CSP. If a CSP consists of a single participant, the designated router for that participant becomes the distribution center.

A center location mechanism should be fast, scalable, and the resultant tree should minimize the consumption of network resources. The proposed mechanism implements a tournament among selected network routers, the winner of which is determined to be the selected center for the center-specific tree. The mechanism relies on the scheduler to assign a *CSP id* to each member of the CSP and a *participant id* to each participant in the interaction. The designated routers for participants with the same *CSP id* enter into a pairwise selection process that results in the selection of a distribution center. The scheduler is responsible for the initial pairing of routers in this process. The initial pairing of routers is based on inter-participant distance since this information is available to the scheduler *a priori*. However, the locations in the subsequent phases are not known *a priori*, and, therefore, these pairings happen without any distance information in this mechanism.

The initial pairing proceeds as follows. Senders are paired with receivers until no more senders or receivers remain. If there are more senders than receivers, any remaining senders are paired with each other (or byes which may be required to make the number of teams in the tournament a power of 2). Otherwise, the remaining receivers are paired together (or with byes). An example tournament can be found in Figure 5 when the center selection protocol is described in section 3.1. Participants that are farthest apart from each other are paired together in order to rapidly move the center towards a cluster of participants. This minimizes the impact of distant participants on the final center selected. Using the same argument we want to minimize the number of successive byes that may occur. If a participant receives a bye in more than a single successive phase, that participant will have a greater impact on the final center selection.

For each pair, a router is selected (called the winner) that represents the middle of the shortest path between the pair. The process is repeated for pairs of winners until a single router remains. The router selected as the winner for the CSP informs the scheduler of the *CSP id* and *group id* for which it won. The scheduler maintains a map of *group ids, CSP ids*, and their associated centers. When all winners have reported, the scheduler sends the registered participants a complete list of center locations for all CSPs in that group. The protocol for the automatic location of distribution centers is given in section 3.1

2.3 TREE CONSTRUCTION

Once the distribution centers are located the participants must become aware of the centers and join the tree. For a participant with a particular *CSP id*, the center with the same *CSP id* in the list supplied by the scheduler is selected as the home center. Each receiver joins all distribution centers to receive traffic, each sender need only join the home center to distribute traffic.

In current techniques, a new group member not connected to a router that is group-aware relies on some protocol like IGMP [13] to reach a router that is group-aware. In the future, we expect that the new member could rely on some hierarchical global group membership service that associates a group name with the address of the nearest distribution center.

The join process is similar to the recently proposed Core Based Trees (CBT) approach [1] with some slight differences due to the asymmetry. A sender joins a distribution center by propagating a join-request along the shortest path to the distribution center. When the join-request reaches the center, a join-ACK is sent back along the same path. A receiver joins in a similar manner by propagating a join-request to the center along the shortest path. The join-ACK is then sent back to the receiver along the shortest path (likely to be different than the path taken by the join-request). Note the above approach allows separate paths for send and receive traffic which could result in routing loops if the mechanism does not guard against it. This is illustrated in Figure 1. The solid arrows represent the paths for send traffic, the dashed arrows represent the paths for receive traffic. Note that at router 1, traffic from sender S1 is forwarded out interfaces a, b, and c but traffic from any other source should only be forwarded out interfaces a and b. To forward S2's data out c would generate a routing loop.

To accommodate this situation, the on-tree routers must maintain a source list. This source list contains the set of interfaces, per sender, that traffic, originating from that sender, should take. This explicit source list increases the state maintained per network node when the number of senders per CSP is large. It should also be noted that the resulting directed graph (as in Figure 1) in no way adheres to the graph theory definition of a "tree." However, throughout the report this graph will still be referred to as a multicast "tree." A detailed description of the tree construction protocol is given in section 3.2

Dynamic membership is handled in a manner similar to CBT. A router is either on-tree or off-tree with respect to a given center-specific tree. A router that is on-tree for any center-specific tree is considered *group aware.* This router maintains a listing of all centers for a particular *group id.*

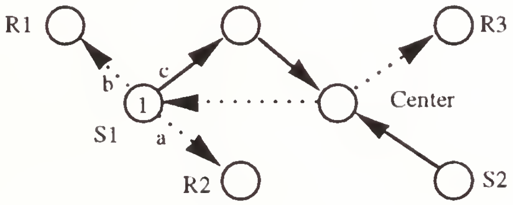


Figure 1. Tree Construction

Currently, no provision is made to relocate distribution centers during an interaction. It is assumed that the number of unanticipated senders in a multiparty interaction does not change excessively throughout the interaction. If this is not the case, a method of periodically relocating the centers is probably sufficient to handle unanticipated participants.

2.4 RESOURCE RESERVATION

In the ideal case, resource reservation should occur before the participants send or receive traffic on the tree. The center for the tree is located based on available resources if the pairwise selection process determines the shortest unicast path using the *unreserved bandwidth* type-of-service (ToS) option such as the one permitted by Open Shortest Path First (OSPF) [6]. Thus, the winner in each phase is selected along the path with the most unused bandwidth available. Use of ToS-based routing in this stage will make the center selection mechanism sensitive to the current network load. Similarly, if the join-requests and join-ACKs are unicast to the center using the same routing option, the branches that are grafted to the tree will contain links with the most bandwidth available. In this manner, the tree is built with resource availability as a primary consideration.

Reservations can occur when the control messages that graft a participant to the tree are sent. Senders are added to the tree via the join-request messages. Reservations can occur when the state for that sender is modified at any router. For receivers, reservations to receive traffic from all current senders can be made as the join-ACK propagates back to the receiver. Thus, the burden of reservation is shared by both senders and receivers. All reservations for a participant are made before the participant sends or receives traffic.

If sufficient resources are unavailable at the time the connection to the tree is to be made, an unreserved connection should take place. This may result in degraded service but minimizes establishment latency over an approach that waits for sufficient resources before the connection is made.

3. PROTOCOLS

Several protocols are required to support the approach. A *participant registration protocol* is required that allows participants to register their attributes with the scheduler before an interaction. A *center selection protocol* is required to implement the pairwise center selection process. A *tree construction protocol* is required to add the senders and receivers to a pre-determined center. A *reservation protocol* is required to obtain reservations on the links of the tree.

A qualitative description of the services provided by each has been given in the previous section. A detailed description of the center selection protocol is presented in section 3.1. A detailed description of the tree construction protocol is presented in section 3.2. The participant registration and reservation protocols will be described in detail in future work.

3.1 CENTER SELECTION PROTOCOL

This section expands on the general description of the center selection mechanism presented in section 2.2. The algorithms for determining the selection hierarchy and the bye positions in the first phase, described in section 2.2, are shown in Figures 2 and 3 respectively [23]. For each pair of participants a winner is determined by the following actions. If the pair consists of a sender and a receiver, the sender is designated as the leader of the pair and a probe is sent via a unreserved bandwidth *type-of-service* unicast along the shortest path to the receiver. This probe is echoed back to the leader along the same path and the route is recorded. From this information the leader selects the middle router along this path as the winner for the pair. Some modifications are necessary if two senders or two receivers are paired together. The participant with the lowest CSP id is designated as the leader. This participant sends a probe along the shortest path to the partner recording the route along the way. The partner then encapsulates this

```
SelectionHierarchy for CSP id = $cspid$ at scheduler
        $numrps$ = number of registered participants in $cspid$;
        number of phases $n = [\log_2 numrps]$;
        /* $slots[i][j]$ is the $j$th winner in phase $i$;*/
        initialize $slots[0][j]\ \forall j \in [1, 2^n]$ with $bye$ using ByeDetermination;
        initialize pairs of empty $slots[0][j]$ with sender-receiver pairs in decreasing order of distance;
        initialize remaining $slots[0][j]$ with remaining participants;
        for $i = 1$ to $n$
                $numslots = 2^{n-i}$ /* number of slots in phase $i$ */;
                for $j = 1$ to $numslots$
                        $slots[i][j]$ = winner of $slot[i-1][2j-1]$ and $slot[i-1][2j]$;
end SelectionHierarchy
```

Figure 2. Algorithm for Determining Selection Hierarchy

information in a new probe which is sent back to the leader along the shortest path (likely a different path altogether). The probe records the route taken. The leader then calculates the cost of the two paths and chooses the middle router along the lower cost path as the winner. For the pairing of the subsequent winners, two probes are always used to determine the lower cost path along which to select a winner. This process can best be illustrated by example.

```
ByeDetermination   by scheduler
        number of phases  $n = [\log_2 numrps]$;
        $byecount = 2^n - numrps$;
        $count = 0$;
        while $byecount > 0$
                $m = 2^{[\log 2 (byecount)]}$;
                for $i = 1$ to $m$;
                        set $slots[0][2^{n - count} - m + i]$ as a bye position;
                $byecount = byecount - m$ ;
                $count = count + 1$;
end ByeDetermination.
```

Figure 3. Determining the Bye Positions in the First Phase

Consider the simple topology in Figure 4a with the two senders and four receivers. The links are bi-directional with asymmetric costs. Figure 5 shows the initial selection hierarchy determined by the scheduler. Each participant knows this hierarchy and from it can determine its partner and the pair of participants whose winner its winner will be paired with. The distance from S1 to R2 (10) is the greatest so this pairing is selected over all others. The next sender-receiver pairing corresponding to the next

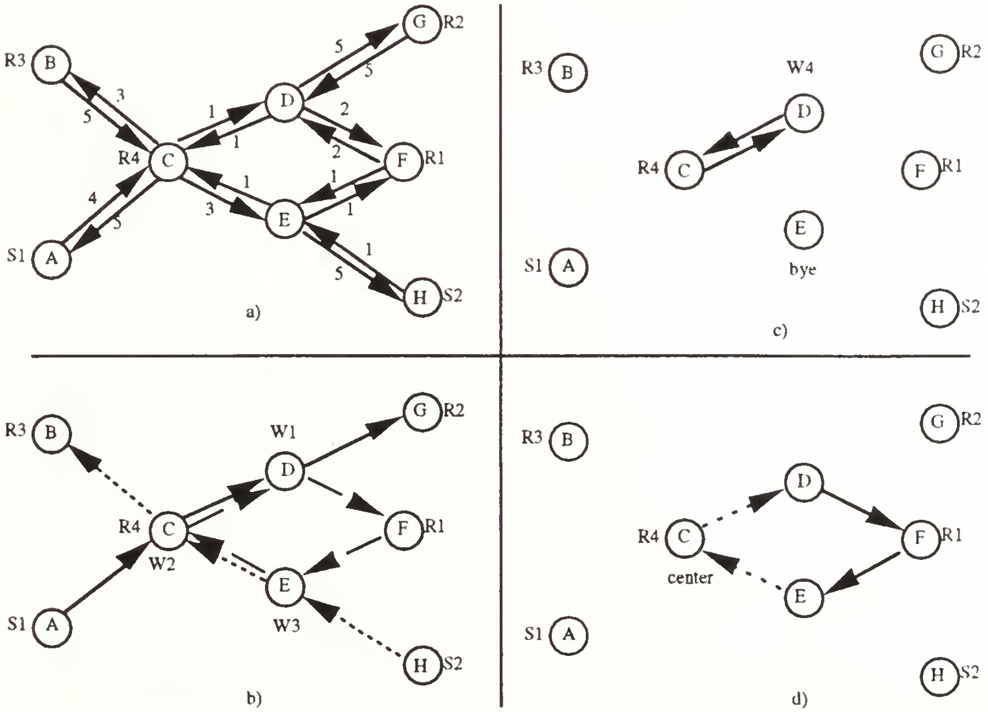


Figure 4. Center Selection Protocol Phases

greatest distance (5) is S2 and R3. This leaves R1 to be paired with R4. The probes of the first phase are shown in Figure 4b. Each pairing is shown with a different arrow type to distinguish it from the other pairs. S1 (router A) sends a probe along the shortest path to R2 (router G). Along this path router D is selected as the winner and designated with the label W1. Likewise, for the pairing of S2 and R3, router C is determined to be the midpoint of the path and is selected as the winner (W2). The pairing of R1 with R4 requires an additional

probe. R1 is designated the leader and sends a probe to R4 via router E. R4 encapsulates this path in a probe that is sent back to R1 along the shortest path. This path takes the probe through router D. The cost of the first path (2) is compared with the cost of the second path (3) and router E is selected as the winner (W3). The second phase is shown in Figure 4c. The winner between routers C and D is determined to be D while router E receives a bye in this phase. The final phase is shown in Figure 4d. In this phase the lower cost path is from router E to D (at a cost of 2) and router C is selected as the overall winner.

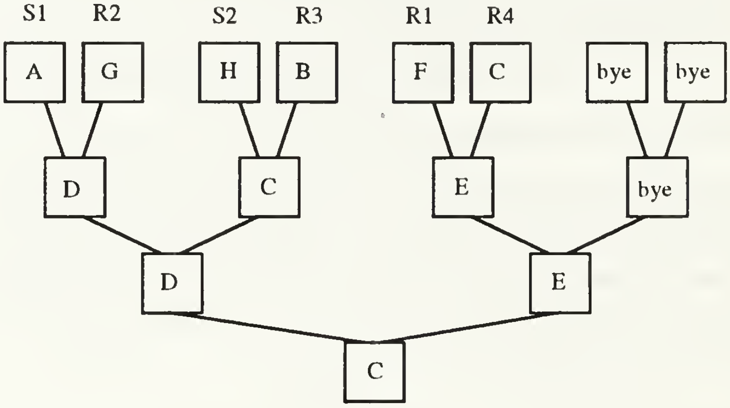


Figure 5. Selection Hierarchy

The method for determining the location of a partner is as follows. The leader of a pair informs all other leaders of the location of its winner. Since the function of all leaders is the same, every leader has a complete list of winners when the phase is over. The leader then informs its winner of the locations of all other winners along with the phase number, the CSP id, the group id, and the leader's id. So after each phase every winner knows every other winner and every winner can determine its partner in the next phase. Since the number of phases required is deterministic ($\log_2$ *numrps*) the winner of the final phase can determine that it is the overall winner and communicate its location to the scheduler along with the *CSP id* it represents.

3.2 TREE CONSTRUCTION PROTOCOL

The method of tree construction presented in section 2.3 requires a protocol to build the state in each router associated with new senders or receivers. This section details such a protocol.

The information, per sender, maintained at each router is [Sender ID, incoming interface, {set of outgoing interfaces (OGI)}]. A group and CSP id should also be attached per sender list. The protocol for adding new senders is presented in Figure 6.

```
1. Propagate the join-request to the center.

2. At each off-tree router along the path:
   Create a sender list (SL) with the new sender that contains the last hop
   as the incoming interface and the next hop as the single OGI.

3. At each on-tree router along the path:
   3.1 Add the sender to the current SL with the incoming interface = last
   hop and {OGIs} = to all OGIs for all other senders in the SL.

   3.2 Send a Build-State message out all OGIs except the OGI that
   represents the next hop to the center.
      3.2.1 If the Build-State encounters a router that does not contain the
      sender in the SL, add the sender to the SL with the incoming
      interface = the last hop, and the {OGIs} = to all OGIs for all
      other senders in the SL.
         3.2.1.1 Send a Build-State message out each of these interfaces.
      3.2.2 If the Build-State encounters a router with an entry for the
       sender in the SL:
            Prune the OGI that led to that router from the {OGIs} for the
            sender at the previous router. If the {OGIs} = { }, remove
            the sender from the SL and and prune the OGI that led to that
            router from the previous router.

            Continue pruning until an entry for the sender is encountered
            that still has at least one other OGI left. The prune message
            knows the location of the previous router from the incoming
            interface entry stored with the sender.
```

Figure 6. Protocol to Add New Senders

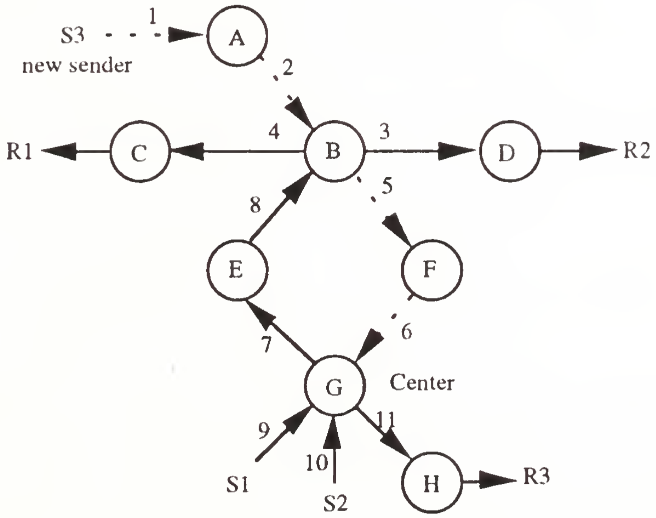


Figure 7. New Sender Example

This protocol is illustrated with the following example. Consider the topology in Figure 7. The dashed arrows represent the path of the join-request for a new sender (S3) to the center (router G). Build-State messages are generated at routers B and G and a prune must occur along the path [B, E, G]. Table 1. shows the state of the tree before the join-request, after all Build-State messages have reached their destinations, and after pruning is complete.

Figure 7 illustrates why the Build-State messages must be sent at each on-tree router. If the Build-State messages were sent only when the join-request reached the center, the Build-State message that reached router B would encounter S3 in the source list and the message would not get propagated out interfaces 3 or 4. This is eliminated if Build-State messages are generated at router B as well as the center.

The protocol for adding new receivers is presented in Figure 8. This protocol is illustrated with the following example. Consider the topology in Figure 9. A new receiver located at router F is to be joined to the tree. The join-request propagates to the center and the join-ACK proceeds to the receiver along the path [A, B, C, D, E, F]. The current state at each router is given in Table 2. As the join-ACK propagates towards the receiver, the state at each router is changed to the New State given in Table 2. The state of the Modified Sender List (MSL) is also shown in this column. When the join-ACK reaches router D it encounters S1 in the sender list (SL) for that router and S1 is also in the MSL. Thus, for S1 a shorter path to router D exists and S1 must be pruned from the path traversed so far. The state of any router that is different as a result of the prune message is presented in the last column of Table 2.

Router	Current State	After Build-State Messages	Pruned State
A	No Senders	S3, 1, {2}	

B	S1, 8, {3, 4}
	S2, 8, {3, 4}
		S1, 8, {3, 4}
		S2, 8, {3, 4}
		S3, 2, {3, 4, 5}	

C	S1, 4, {R1}
	S2, 4, {R1}
		S1, 4, {R1}
		S2, 4, {R1}
		S3, 4, {R1}	

D	S1, 3, {R2}
	S2, 3, {R2}
		S1, 3, {R2}
		S2, 3, {R2}
		S3, 3, {R2}	

E	S1, 7, {8}
	S2, 7, {8}
		S1, 7, {8}
		S2, 7, {8}
		S3, 7, {8}
			S1, 7, {8}
			S2, 7, {8}
			Remove S3

F	No Senders	S3, 5, {6}	

G	S1, 9, {7, 11}
	S2, 10, {7, 11}
		S1, 9, {7, 11}
		S2, 10, {7, 11}
		S3, 6, {7, 11}
			S1, 9, {7, 11}
			S2, 10, {7, 11}
			S3, 6, {11}

H	S1, 11, {R3}
	S2, 11, {R3}
		S1, 11, {R3}
		S2, 11, {R3}
		S3, 11, {R3}	

Table 1. Adding a New Sender

Some additional functions are required in the join-ACK to accommodate the pruning process. In order to determine the path along which to send the prune message, the join-ACK must record its route as it propagates to the receiver or this path must be discerned by the prune message from the state information at each router (i.e. using the incoming and outgoing interfaces from the sender list. When the prune message

```
From the center, propagate the join-ACK to the receiver.

1. For any sender at the center with the next hop not in {OGIs}, add the next
hop to the {OGIs}. Add these senders to the Modified Sender List (MSL) in
the join-ACK.

2. Proceed to the Receiver

3. At each hop:
    3.1 If a sender in the MSL is NOT a member of the SL at the hop:
        Add the sender to the SL with the last hop as the incoming interface
        and the next hop as the single OGI.

    3.2 If a sender that is NOT in the MSL is encountered in a SL:
        Check the senders {OGIs} for the next hop:
            If next hop in {OGIs}, do nothing
            If next hop NOT in {OGIs}, add next hop to {OGIs} and add the
                sender to the MSL.

    3.3 If a sender that is in the MSL is encountered in a sender list:
            Prune the previous path of the join-ACK by removing the sender
            from any router for which |OGI|=1. When |OGI| >1 for that
            sender, or the center is reached, remove the OGI corresponding to
            the next hop for the receiver from the sender's {OGIs}.

            Check the encountered sender's {OGIs}:
                If the next hop ∈ {OGIs}, remove sender from MSL.
                If the next hop ∉ {OGIs}, add next hop to {OGIs}.
```

Figure 8. Protocol to Add New Receivers

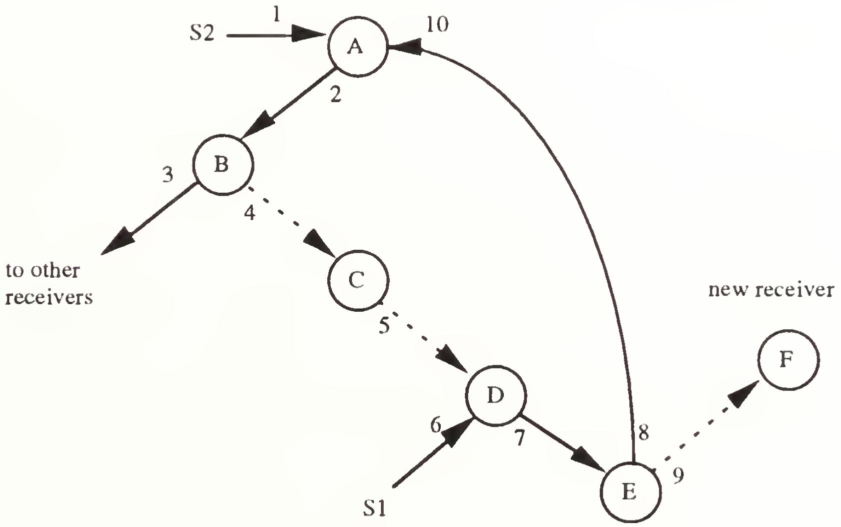


Figure 9. New Receiver Example

eventually encounters a set of OGIs of size greater than one, it must determine which interface to prune from this set. Given the identity of the receiver, the interface corresponding to the next hop for the receiver should be the interface that gets pruned. Thus, the prune message must know the sender id that it is pruning and the receiver id it is pruning that sender for.

The first participant that joins the tree is critical to the correct formation of the tree. If a sender is the first to join, the tree will be built correctly. However, if a receiver joins before any other sender, then there is no state at the center to propagate back towards the receiver. To remedy this, the center can either wait for a sender to join before propagating the join-ACK or the join-ACK can be sent with the center listed as a sender. This second case allows a trail to be built to the receiver that a new sender can follow. No traffic will be sourced from the center and its entry in each SL requires minimal overhead.

Router	Current State	New State	Pruned State

A	S1, 10, {1,2}
	S2, 1, {2}
		No Change
		MSL = { }	

B	S1, 2, {3}
	S2, 2, {3}
		S1, 2, {3, 4}
		S2, 2, {3, 4}
		MSL={S1, S2}
			S1, 2, {3}
			S2, 2, {3,4}

C	No Senders	S1, 4, {5}
		S2, 4, {5}
		MSL={S1, S2}
			Remove S1
			S2, 4, {5}

D	S1, 6, {7}	S1, 6, {7}
		S2, 5, {7}
		MSL={S2}
			Generate Prune

E	S1, 7, {8}	S1, 7, {8, 9}
		S2, 7, {9}
		MSL={S1, S2}	

F	No Senders	S1, 9, {Rec}
		S2, 9, {Rec}	

Table 2. Adding a New Receiver

4. SIMULATION RESULTS

This section describes the evaluation of the center selection and tree construction techniques detailed in the previous sections. The objective is to compare the tree cost and average path length as the number of concurrent senders is varied for two types of trees. The first is a center-specific shortest path shared tree with the center located according to the proposed approach. The second is a source-specific shortest path tree. The ground work for the following simulations can be found in [23] and the results further support the preliminary findings detailed in that work.

4.1 ENVIRONMENT

A random network is generated using Waxman's RG1 and RG2 [9] algorithms. Several clusters (meant to simulate separate routing domains) are generated using RG2 and these clusters are connected by links generated using either RG1 or RG2. A maximum cost between any two nodes (L) and the total number of nodes (N) is established. For RG2, every pair of nodes (i,j) is assigned an integer cost ($d_{i,j}$) utilizing a uniform distribution from 1 to L. For RG1, the nodes are scattered in an NxN matrix and the Euclidean distance ($d_{i,j}$) between any node pair (i,j) is calculated. The probability ($p_{i,j}$) that a link exists between a node pair is given by $p_{i,j} = \beta\, e^{-d_{i,j}/(\alpha L)}$. If a link exists, its cost is $d_{i,j}$. If link (i,j) exists then $p_{j,i} = 1$ but the cost $d_{j,i}$ is completely independent of cost $d_{i,j}$. The parameters α and β are defined on the interval (0,1]. A small value of α will cause a relatively greater number of low cost links. The node degree, (λ_i), (the number of links from a given node) for node *i* is approximated by

$$\lambda_i \approx \sum_{j=1, j\neq i}^{N} p_{i,j} = \beta \sum_{j=1, j\neq i}^{N} e^{-d_{i,j}/\alpha L}$$

Since any cost ($d_{i,j}$) inside a cluster is a uniformly distributed integer between [1, L] there will be approximately (N - 1)/L links of each possible value of $d_{i,j}$. This allows the following revision.

$$\lambda_i \approx \frac{\beta(N-1)}{L} \sum_{k=1}^{L} e^{-k/\alpha L}$$

Letting $e^{-1/(\alpha L)} = \rho$ and observing that the above equation is a finite geometric sum of ρ yields

$$\lambda_i \approx \frac{\beta(N-1)\,\rho(1-\rho^L)}{L(1-\rho)}$$

Since λ_i can be approximated by this method for every i, the average node degree λ_{avg} for the entire graph can be approximated by this formula.

$$\lambda_{avg} \approx \frac{\beta(N-1)\,\rho(1-\rho^L)}{L(1-\rho)}$$

Solving for β results in the following

$$\beta \approx \frac{\lambda_{avg}\, L(1-\rho)}{(N-1)\,\rho(1-\rho^L)}$$

If further simplification is desired ρ^L can be approximated as 0 and N - 1 can be approximated as N for L >> 1 and N >> 1 respectively.

The tree cost for each tree is calculated by determining the number of senders that use each link. Let $S_{i,j}$ be the number of senders that use the link from node *i* to node *j* and $d_{i,j}$ be the cost to use that link. Let *ss* be the number of concurrent senders. The tree cost for that link, $tc_{i,j}$, is given by $tc_{i,j} = \min(ss, S_{i,j})d_{i,j}$. The total tree cost, tc_{tot}, is given by

$$tc_{tot} = \sum_{i=1}^{N} \sum_{j=1}^{N} tc_{i,j}$$

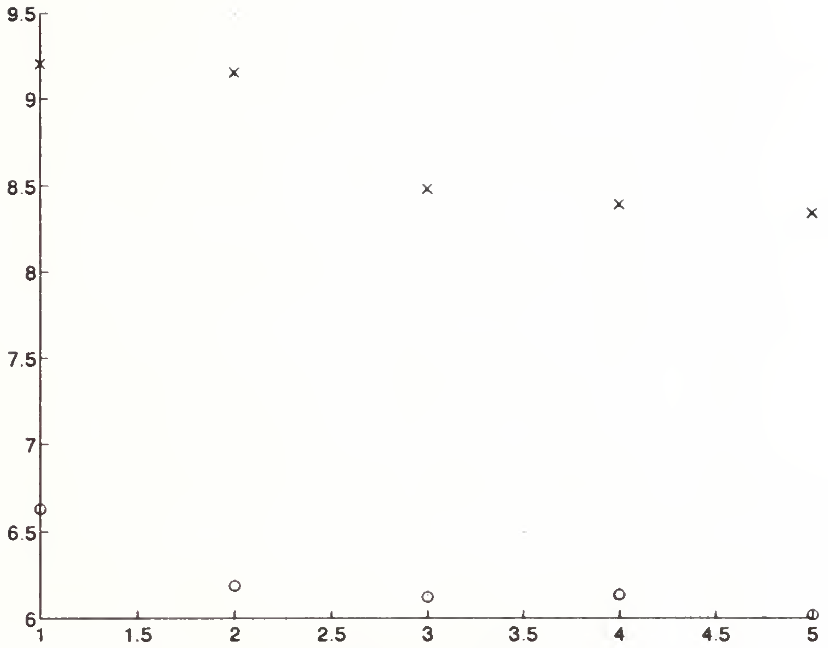
For the center-specific trees, separate send and receive paths are allowed as per the tree construction protocol. For both tree types the average path length is computed as the average number of hops experienced by a sender to all receivers averaged over all senders.

4.2 RESULTS

Simulations were carried out to determine how the center-specific trees compare with source-specific trees as the number of concurrent senders in an interaction increased. The topologies varied from 10-100 nodes distributed over 1 to 10 clusters. Link cost was defined as a uniform random variable between 1-100 for intra-cluster links and 1-1000 for inter-cluster links. Node degree was kept in the interval [3,5] by manipulating α and β for each topology. The simulations allowed multiple interactions to be constructed on top of each other. As one interaction was built (each interaction is referred to as an "iteration" of the approach) the network resources required by that interaction were consumed in the state of available bandwidth and additional sessions were then built using the new bandwidth state. Thus, separate bandwidth state was maintained for each tree type to facilitate comparison of the tree types in the presence of multiple interactions.

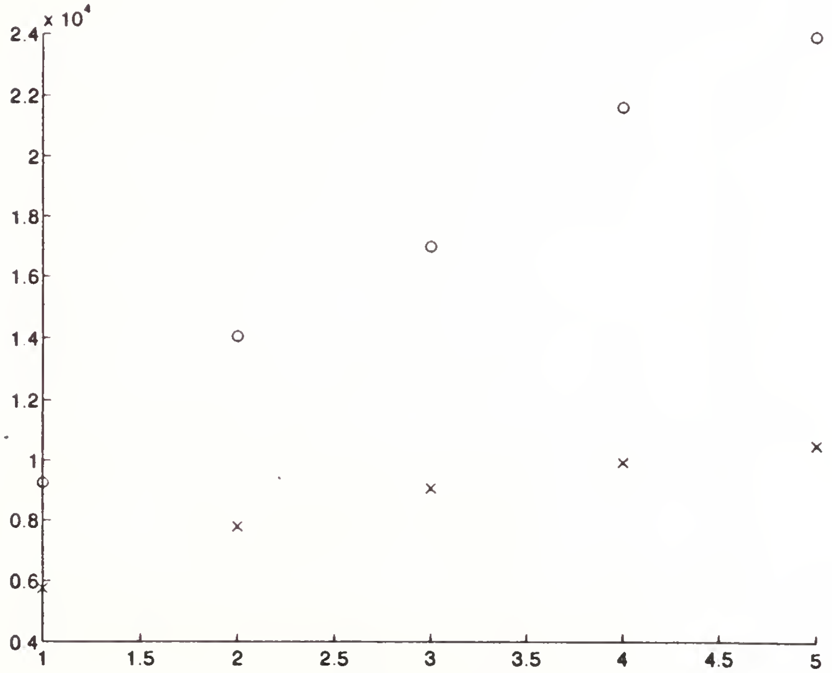
Figures 8, 9, and 10 show a representative sample of results generated through extensive simulations. These particular results are for a connected topology with three clusters each with 30 nodes. The cost of the inter domain links is, on average, an order of magnitude greater than the intra domain links. The results cover five iterations each with 12 senders and 12 receivers evenly distributed over all domains. The first results (Figures 8-11) are given for a topology with an average node degree of 3. Figure 8 shows the average path length (in number of hops) experienced by each sender per iteration. This path length is proportional to the delay the multicast traffic experiences on the tree. Figure 9 shows the tree cost per iteration for 3 simultaneous senders. By observing the slope of the two plots, it is evident that the proposed approach distributes network load better than source-specific trees over several iterations. Figure 10 shows the tree cost for both trees as the number of concurrent senders is increased (up to the total number of senders defined in the simulation) In Figure 10, the source-specific tree starts out with a higher cost, peaks more rapidly, but levels off as the number of concurrent senders increases. The center-specific tree tends to increase at a constant rate. When the number of concurrent senders is large it is possible for the cost of the center-specific tree to exceed that of the source-specific tree. The shape of the source-specific tree is due to a greater number of shared links, but these links are shared by only a few senders, whereas the links of the center-specific tree are shared by most senders.

The center selection algorithm was evaluated by comparing the cost of the center-specific tree against the cost of the "send" and "receive" tree rooted at the optimal send and receive centers. A "send" tree

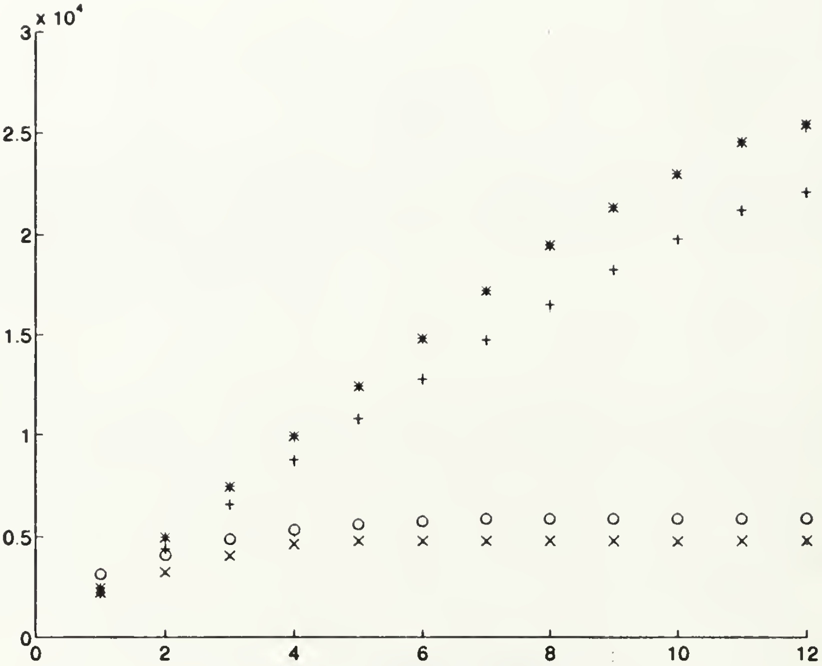


Number of Interactions
x = Center-Specific Tree, o = Source-Specific Tree

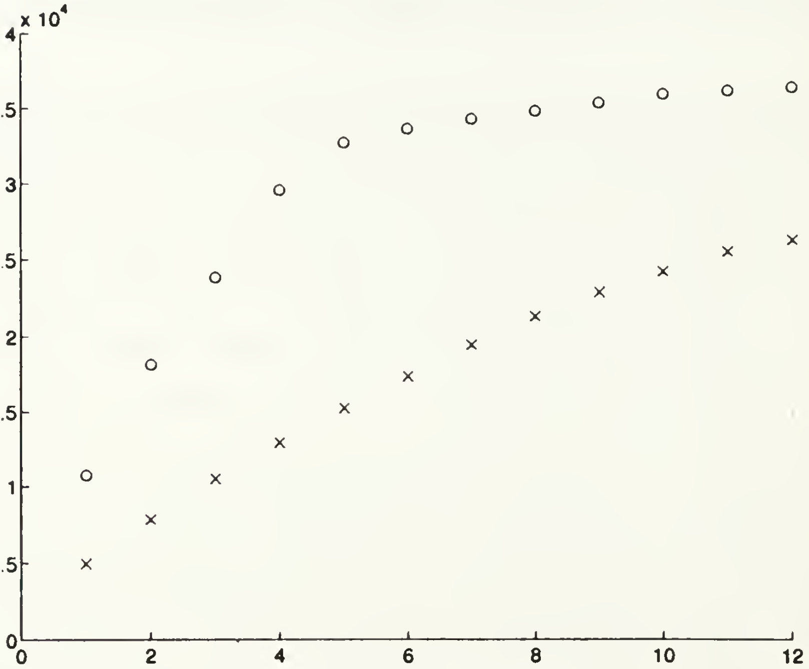
Figure 8. Average Path Length per Interaction



Number of Interactions
x = Center-Specific Tree, o = Source-Specific Tree
Figure 9. Tree Cost per Interaction with 25% of Senders Active



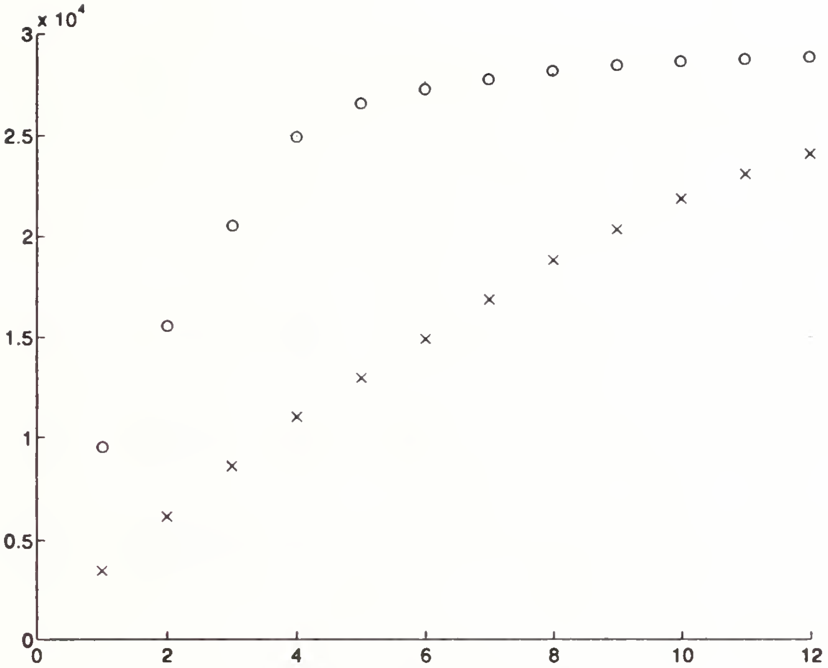
Number of Simultaneous Senders
x = Center-Specific Tree, o = Source-Specific Tree
Figure 10. Tree Cost vs. Simultaneous Senders on Final Interaction



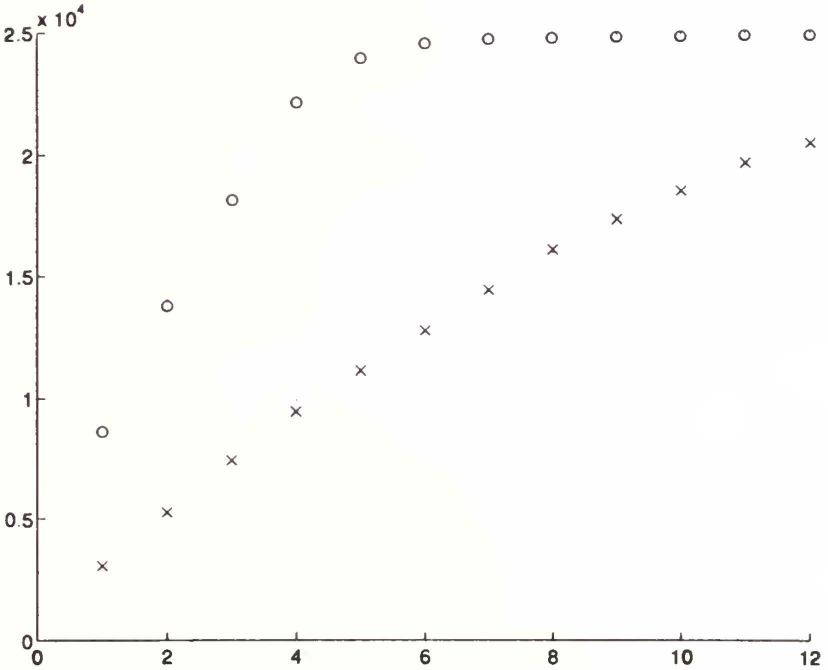
Number of Simultaneous Senders

* = Tree Cost for Receive Tree Rooted at Selected Center
o = Tree Cost for Send Tree Rooted at Selected Center
+ = Tree Cost for Receive Tree Rooted at Least Cost Center
x = Tree Cost for Send Tree Rooted at Least Cost Center

Figure 11. Evaluation of Center Selection Protocol
With Respect to the Number of Simultaneous Senders



Number of Simultaneous Senders
x = Center-Specific Tree, o = Source-Specific Tree
Figure 12. Tree Cost vs. Simultaneous Senders for Node Degree = 4



Number of Simultaneous Senders
x = Center-Specific Tree, o = Source-Specific Tree
Figure 13. Tree Cost vs. Simultaneous Senders for Node Degree = 5

was computed as the one-way tree from all senders to a send center. Likewise, a "receive" tree was computed as the one-way tree from a receive center to all receivers. Optimal send centers were located as follows. For every router in the network, an average least cost from all senders to that router was computed. The router representing the least average cost was selected as the optimal send center. A similar computation was performed to locate the optimal receive center.

Figure 11 shows a plot of the tree cost for the send and receive trees with the one-way tree cost for the proposed center-specific tree imposed on top of these costs. The costs are shown as the number of concurrent senders is increased. From these tests it is evident that the proposed approach locates centers that represent a good compromise between the sending and receiving requirements of the interaction.

A critical element of any topology is the node degree. Figures 12 and 13 show how the two trees behave as the node degree is increased. As the node degree increases it is evident that the cost of source-specific trees peaks more quickly and the crossover point for center-specific trees is achieved sooner since the relative distance between the two curves is smaller.

4.3 SUMMARY OF SIMULATION RESULTS

The main conclusion to be drawn from the simulations is that the center-specific trees do provide a lower tree cost than source-specific trees without sacrificing significantly in delay even for multiple concurrent senders. Source-specific trees are more economical than center-specific trees when the number of concurrent senders is large and when the network is highly interconnected. The cross-over point depends upon both the richness of the topology and the number of concurrent senders.

The simple center location protocol is very effective in generating low cost center-specific trees. It performs even better in topologies that are not highly interconnected. It must be noted that our results are significant if multiparticipant interactions are to be set up with reservation of resources to guarantee QoS. When reservations are required, each tree cost calculated above can be interpreted as the total number of resources to be reserved in the network to support the corresponding number of concurrent senders.

5. RELATED WORK

In this section we examine the extent to which the existing protocols meet the following design goals previously identified for network level multicast with guaranteed QoS. The design goals are:

1. Use of *a priori* information about participants,
2. Automatic location of distribution centers,
3. Flexible selection between source specific and center-specific trees,
4. Tree formation based on resource availability,
5. Support of multiple routing protocols,
6. Minimal tree state information,
7. Minimal per-packet processing in the routers,

8. Participation of senders as well as receivers in reservation.

9. Minimal resource consumption in asymmetric network topologies.

Note, goals 6 and 7 are somewhat conflicting. Table 1 summarizes the comparison among existing protocols. The Distance Vector Multicast Routing Protocol (DVMRP) [5] and Multicast Extensions to OSPF (MOSPF) [7] automatically locate a center since they utilize source-specific trees only. These protocols are only applied to intra-domain multicast routing and therefore are not required to support multiple protocols. Goal 8 is not listed in the table since none of the existing protocols supports reservations. The completely independent RSVP implements receiver based reservation.

MOSPF is the only existing technique that handles asymmetric network topologies. Since the topological database in MOSPF is stored as a directed graph, the link costs are bi-directional and the Dijkstra shortest path trees are computed using this state information. Techniques using reverse path multicasting (RPM) or some variant of RPM obviously suffer when the link costs are asymmetric [15]. Thus, techniques such as Dense Mode PIM and DVMRP do not support asymmetric topologies. Sparse mode PIM sets up the packet delivery path as PIM-join messages propagate towards the RP or source for each receiver. Assuming that the path taken by the PIM-join is the shortest path to the RP or source, it may not represent the shortest path that the actual traffic to the receiver must take if the link costs are asymmetric. Thus, the resultant shared or source based trees for Sparse Mode PIM may be sub-optimal. Tree construction for CBT also suffers from a similar phenomenon when the network topology is asymmetric. For ToS based routing the designers of PIM suggest in [2] that a symmetry flag be used by BGP/IDRP [17, 18] that allows PIM to determine if packets will be allowed to travel in the reverse

Goal	CBT [1]	PIM [2,3,4]	DVMRP [5, 8]	MOSPF [7, 8]	Proposed Approach
Use of *a priori* information	N	N	N	N	Y
Automatic location of distribution centers	N	N	Y	Y	Y
Choice between SST and CST	N	Y	N	N	Y
Tree formation based on resource availability	N	N	N	N	Y
Minimal router processing	Y	Note 1	N	N	Y
Minimal tree state information	Y	Note 2	Y	N	Note 3
Support of multiple routing protocols	N	Y	N	Y	Y
Support for asymmetric topologies	N	N	N	Y	Y

Note 1: Y for sparse mode, N for dense mode
Note 2: N for sparse mode, Y for dense mode
Note 3: Moderate amount of state, less than PIM but more than CBT

Table 3: Existing Approaches vs. Design Goals

direction. If this flag is not set PIM suggests looking for an SDRP [19] route that has the flag set. This requires SDRP to carry the symmetry flag and that PIM messages follow an SDRP route. This approach appears to select arbitrary paths relative to QoS requirements and then hopes that the paths are symmetric by checking the symmetry flag. Additional questions remain concerning conditions under which the symmetry flag will be set.

The tree state information required at each on-tree router in the approach proposed in this report is of the same order as the state information required by MOSPF or Sparse Mode PIM (identified by the multicast forwarding entry for (S,G) required at each on-tree router). So while the proposed approach may not meet goal 6 completely, it meets it to the same extent as some existing techniques.

The proposed approach also compares favorably in size with Sparse Mode PIM. PIM requires that every receiver learn of each sender through an RP. This implies that an RP, must continue to listen to every sender and each sender must continue to send to every RP, even if no receivers are receiving traffic through the RP. Let N_s, N_r, and N_c be the number of senders, receivers, and RPs (in the case of center-specific trees, the number of centers) respectively. In PIM, a new receiver gets all its multicast traffic from a single RP regardless of the number of senders. In the proposed approach each receiver attaches itself to all centers and a sender sends traffic to only one center. A rough estimate of the size of the solution for PIM is $N_s N_C$. This is provided that all receivers get their traffic through an RP. If all receivers opt to gather their traffic from every source, the metric becomes bounded by $N_s N_r + N_s N_c$, a rather significant increase.

PIM requires that the RPs maintain a complete list of senders and that on-tree routers maintain a multicast forwarding entry for the shared tree and for all sources on shortest path trees. The proposed approach requires that a partial list of senders (containing the same information as PIM multicast forwarding entries for (S,G)) be maintained at all on-tree routers. The approach will still scale if the number of senders per CSP and the number of routers shared by more than one tree is small. Some control can be exerted to meet the first condition, but routers near the receivers will likely be members of all trees. If R_i = {routers on-tree for CSP *i*} and S_i = {senders for CSP *i*} then a rough estimate for the size of the proposed approach is:

$$\sum_{i=1}^{n} | S_i | | R_i |$$

where n is the number of CSPs (equal to N_c in the discussion above).

6. CONCLUDING REMARKS

This report addressed the problem of constructing multicast trees with guaranteed QoS that utilize network resources efficiently. It identified design goals for constructing such trees and presented an integrated approach to achieve an efficient combination of center-specific trees and source-specific trees based on *a priori* information about participants. It describes a scalable center location mechanism and protocol for locating a distribution center that balances network resource consumption. It describes an approach and protocol for constructing a center-specific tree around a pre-selected distribution center in a network with asymmetric link costs. It is shown, by simulation modeling, that the resultant center-specific trees are efficient in the presence of multiple concurrent senders in terms of delay and resources consumed.

Since the state information required by the proposed approach introduces some complexity, some alternatives should be examined. The simplest alternative is to force the same path for traffic in both directions. While this will result in the degradation of some traffic (since the path it must traverse is no longer the shortest) some improvement can still be made over existing approaches. A determination can be made to minimize the degradation suffered by the traffic by picking a path that represents the best compromise between the two directions. While the trees in the proposed approach will consume fewer resources, the state information in each router is no longer required. Since the overall goal was to minimize the network resources consumed by the tree (to guarantee QoS), separate send and receive paths are proposed at the expense of minimal tree state information.

The two main contributions of this work were motivated by the idea of proposing enhancements to existing protocols. The center selection mechanism was motivated by the need to determine a location for cores in a CBT approach or RP's in a Sparse Mode PIM approach. The proposed tree construction algorithm has its origins in the tree construction phase proposed by CBT with the state information taken from Sparse Mode PIM and MOSPF. While the need for the state information in the proposed approach (separate paths for send and receive traffic) is different than that of Sparse Mode PIM (the superposition of shared and source-based trees) the complexity imposed by each remains the same.

The following issues need to be addressed if an implementation of this approach is to be undertaken:

1. A detailed specification of the tree construction protocol, particularly at the boundary conditions.

2. A detailed specification of a protocol to remove state information for departing participants. We expect this protocol to be similar to the tree construction protocol.

3. Analysis of the tree construction protocol under transient conditions (e.g. during simultaneous joins).

4. Formal specifications of the above protocols and associated proofs.

5. Detailed specifications of the registration protocol, and reservation protocol.

6. A method for reconfiguring the distribution centers over the lifetime of an interaction.

The contributions of this work are:

1. Use of *a priori* information about participants to provide multicast data distribution that permits a flexible combination of center-specific and sender-specific trees.

2. An approach to algorithmically locate a center for a center-specific tree.

3. An approach to center-specific tree construction in the presence of network asymmetry.

4. Simulations detailing the quality of the resultant trees.

LIST OF REFERENCES

[1] Tony Ballardie, Paul Francis, and Jon Crowcroft, "Core based trees (CBT) an architecture for scalable inter-domain multicast routing," in *ACM SIGCOMM*, September 1993.

[2] Stephen Deering, Deborah Estrin, Dino Farinacci, Van Jacobson, Ching-Gung Liu, and Liming Wei, *Protocol Independent Multicast (PIM): Motivation and Architecture*, Internet Draft, draft-ietf-idmr-pim-arch-00.ps, March 1994.

[3] Stephen Deering, Deborah Estrin, Dino Farinacci, Van Jacobson, Ching-Gung Liu, and Liming Wei, *Protocol Independent Multicast (PIM), Sparse Mode Protocol Specification,* Internet Draft, draft-ietf-idmr-pim-sparse-spec-00.ps, March 1994.

[4] Stephen Deering, Deborah Estrin, Dino Farinacci, and Van Jacobson, *Protocol Independent Multicast (PIM), Dense Mode Protocol Specification,* Internet Draft, draft-ietf-idmr-pim-dense-spec-00.txt, March 1994.

[5] D. Waitzman, C. Partridge, and S. Deering, *Distance Vector Multicast Routing Protocol,* Technical Report RFC 1075, Internet, Network Working Group, November 1988.

[6] J. Moy, *OSPF Version 2*, Technical Report RFC 1247, Internet, Network Working Group, July 1991.

[7] J. Moy, *Multicast Extensions to OSPF*, Technical Report RFC 1584, Network Working Group, March 1994.

[8] Stephen Deering and David Cheriton, "Multicast Routing in Datagram Internetworks and Extended LANs," *ACM Transactions on Computer Systems,* May 1990.

[9] Bernard M. Waxman, "Routing of Multipoint Connections," *IEEE Selected Areas in Communications,* December 1988.

[10] David D. Clark, Scott Shenker, and Lixia Zhang, "Supporting Real-time Applications in an Integrated Services Packet Network: Architecture and Mechanism," *ACM SIGCOMM,* September 1992.

[11] Stephen Deering, *Multicast Routing in a Datagram Network*, Ph.D. thesis, December 1991.

[12] Liming Wei and Deborah Estrin, *A Comparison of Multicast Trees and Algorithms,* Draft Submitted to INFOCOM 1994, 1993.

[13] Stephen E. Deering, *Host Extensions for IP Multicasting*, Technical Report RFC 1112, Internet, Network Working Group, August 1989.

[14] L. Zhang, R Braden, D Estrin, S Herzog, and S. Jamin, *Resource ReSerVation Protocol (RSVP)-Version 1 Functional Specification,* Internet Draft, draft-ietf-rsvp-spec-03.ps, July 1994.

[15] Robert J. Barton, *Multicast Analysis Simulation Tool (MAST),* Technical Report, Naval Postgraduate School, September 1994.

[16] Van Jacobson, *The Session Directory Tool,* Lawrence Berkeley Laboratory, 1992.

[17] Y. Rekhter and T. Li, editors. *A Border Gateway Protocol 4 (BGP-4),* Internet Draft, January 1994.

[18] S. Hares and John Scudder, *IDRP for IP*, Internet Draft, September 1993.

[19] D. Estrin, T. Li, Y. Rekhter, and D. Zappala, *Source demand Routing Protocol: Packet Format and Forwarding Specification*, Internet Draft, March 1993.

[20] K. Claffy, G. Polyzos, and H. W. Braun, "Traffic Characteristics of the T1 NSFNET Backbone," *Proceedings of INFOCOM '93*, March 1993.

[21] Merit Inc., Statistics available via anonymous ftp to nic.merit.edu, directory nsfnet/statistics; May 1994.

[22] Vern Paxson, "Growth Trends in Wide-Area TCP Connections," *IEEE Network*, August 1994.

[23] Eric Boyer, *Multicast Communication With Guaranteed Quality of Service*, Master's Thesis, Naval Postgraduate School, December 1993.

INITIAL DISTRIBUTION LIST

1. Defense Technical Information Center 2
Cameron Station
Alexandria, VA 22304-6145

2. Dudley Knox Library, Code 52 2
Naval Postgraduate School
Monterey, CA 93943-5101

3. Chairman, Code EC 1
Department of Electrical and Computer Engineering
Naval Postgraduate School
Monterey, CA 93943-5121

4. Professor Shridhar B. Shukla, Code EC/Sh 8
Department of Electrical and Computer Engineering
Naval Postgraduate School
Monterey, CA 93943-5121

5. James Eric Klinker 1
Code 5544
Naval Research Laboratory
4555 Overlook Ave. SW
Washington, D.C. 20375-5000

DUDLEY KNOX LIBRARY
3 2768 00327824 3